LOPS: Learning Order Inspired Pseudo-Label Selection for Weakly Supervised Text Classification

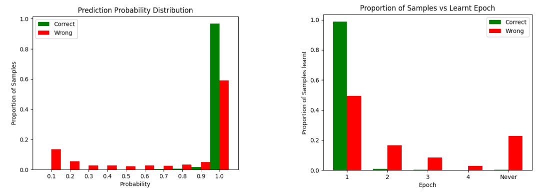
Anonymous ACL submission

Abstract

Weakly supervised text classification methods typically train a deep neural classifier based on pseudo-labels. The quality of pseudo-labels is crucial to final performance but they are inevitably noisy due to their heuristic nature, so selecting the correct ones has a huge potential for performance boost. One straightforward solution is to select samples based on the softmax probability scores in the neural classifier corresponding to their pseudo-labels. However, we show through our experiments that such solutions are ineffective and unstable due to the erroneously high-confidence predictions from poorly calibrated models. Recent studies on the memorization effects of deep neural models suggest that these models first memorize training samples with clean labels and then those with noisy labels. Inspired by this observation, we propose a novel pseudo-label selection method LOPS that takes learning order of samples into consideration. We hypothesize that the learning order reflects the probability of wrong annotation in terms of ranking, and therefore, propose to select the samples that are learnt earlier. LOPS can be viewed as a strong performance-boost plug-in to most of existing weakly-supervised text classification methods, as confirmed in extensive experiments on four real-world datasets.

1 Introduction

Weakly supervised text classification methods (Agichtein and Gravano, 2000; Riloff et al., 2003; Tao et al., 2015; Meng et al., 2018; Mekala and Shang, 2020; Mekala et al., 2020, 2021) typically start with generating pseudo-labels, and train a deep neural classifier to learn the mapping between documents and classes. There is no doubt that the quality of pseudo-labels plays a fundamental role in the final classification accuracy, however, they are inevitably noisy due to their heuristic nature. Pseudo-labels are typically generated by some heuristic, for example, through string matching between the documents and user-provided seed words (Mekala and Shang, 2020). Deep neural networks (DNNs) trained on such noisy labels have a high risk of making erroneous predictions. More importantly, when self-training is employed, such error can be further amplified upon boostrapping.



(a) (b)

Figure 1: Distributions of correct and wrong instances using different pseudo-label selection strategies on the NYT-Coarse dataset for its initial pseudo-labels. The base classifier is BERT. (a) is based on the softmax probability of samples' pseudo-labels and (b) is based on the earliest epochs at which samples are learnt.

To address this problem, in this paper, we study the pseudo-label selection in weakly supervised text classification, aiming to select a high quality subset of the pseudo-labeled documents (in every iteration when using self-training) that can potentially achieve a higher classification accuracy.

A straightforward solution is to first train a deep neural classifier based on the pseudo-labeled documents and then threshold the documents by the predicted probability scores corresponding to their pseudo-labels. However, DNNs usually have a poor calibration and generate overconfident predicted probability scores (Guo et al., 2017). For example, on New York Times (NYT) coarse-grained dataset, as shown in Figure 1(a), 60% of wrong instances in the pseudo-labeled documents have a predicted probability by BERT greater than 0.9 for their wrong pseudo-labels. There are recent works that use uncertainty to fix calibration (Rizve et al., 2021) and other lines of work focusing on label selection from noisy data (Jiang et al., 2018b; Han

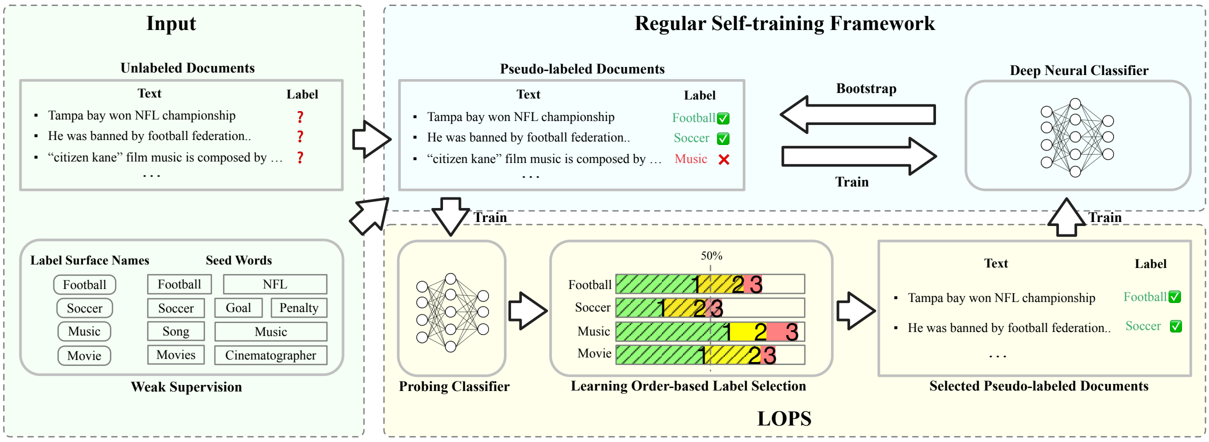


Figure 2: Usually self-training frameworks follow the path in the top block, starting from generating noisy pseudo-labeled documents, training the text classifier, and bootstrapping by adding high confidence predictions. We propose to add a step "Label Selection" (shown in below block) to select the correctly labeled documents. LOPS trains a classifier to obtain the learning order of samples and we stop the training when at least $\tau\%$ of samples corresponding to each class are learnt and select the learnt samples. The numbers shown are learnt epochs and the samples in the shaded part are selected.

et al., 2018; Ren et al., 2018; Fang et al., 2020), however, all these methods require a clean validation set, whereas in our problem, we have no human-annotated documents at all.

Recent studies on the memorization effects of DNNs show that they memorize easy and clean instances first, and gradually learn hard instances and eventually memorize the wrong annotations (Arpit et al., 2017; Geifman et al., 2018; Zhang et al., 2021). We have confirmed this in our experiments for different classifiers. For example, as shown in Figure 1(b), BERT classifier learns most of the clean instances in the first epoch and learns wrong instances across all epochs. Although it also learns good number of wrong instances in the first epoch, it is significantly less than the probability-based selection in Figure 1(a). Since the correct samples are learnt first, we hypothesize that learning order-based selection will be able to filter out the wrongly labeled samples.

Inspired by our observation, we propose a novel learning order inspired pseudo-label selection method LOPS, as shown in Figure 2. Specifically, LOPS involves training a classifier and tracking the learning order of samples. We define a sample is learnt if and only if the classifier trained on pseudo-labels gives the same argmax prediction as its pseudo-label at the end of an epoch. We stop the training when at least $\tau\%$ of samples corresponding to each class are learnt and select all the learnt samples. We empirically show that LOPS can boost the accuracy of various weakly supervised text classification methods and it is much more effective and stable than probability score-based selections.

Our contributions are summarized as follows:

- We propose a novel pseudo-label selection method LOPS that takes learning order of samples into consideration.
- We show that selection based on learning order is much stable and effective than selection based on probability scores.
- Extensive experiments and case studies on real-world datasets with different classifiers and weakly supervised text classification methods demonstrate significant performance gains upon using LOPS. It can be viewed as a solid performance-boost plug-in for weak supervision.

Reproducibility. We will release the code and datasets on Github[1].

2 Related Work

We review the literature about (1) pseudo-labeling in weakly supervised text classification, (2) label selection methods, and (3) training dynamics.

Pseudo-Labels in Weakly Supervised Text Classification. Since the weakly supervised text classification methods lack gold annotations, pseudo-labeling has been a common phenomenon to generate initial supervision. Pseudo-labeling depends on the type of weak supervision. Mekala and Shang (2020) and Mekala et al. (2020) have a few label-

[1] https://github.com/anonymous

indicative seed words as supervision and they generate pseudo-labels using string-matching where a document is assigned a label whose aggregated term frequency of seed words is maximum. (Meng et al., 2018) generates pseudo-documents using the seed information corresponding to a label. (Wang et al., 2020) takes only label names as supervision and generates class-oriented document representations, and cluster them to create a pseudo-training set. Under the same scenario, (Mekala et al., 2021) consider samples that exclusively contain the label surface name as its respective weak supervision. In (Karamanolakis et al., 2021b), pseudo-labels are created from the predictions of a trained neural network. (Arachie and Huang, 2021) combines different weak signals to produce probabilistic training labels All the above mentioned methods involve learning from noisy data and our label selection method substantially reduces the noise and improves their performance.

Label Selection. There are different lines of work aiming to select true-labeled examples from a noisy training set. One line of work involves training multiple networks to guide the learning process. Along this direction, (Malach and Shalev-Shwartz, 2017) maintains two DNNs and update them based on their disagreement. (Jiang et al., 2018b) learns another neural network that provides data-driven curriculum. (Han et al., 2018; Yu et al., 2019) use co-training where they select instances based on small loss criteria and cross-train two networks simultaneously. Another line of work learns weights for the training data. Along this line, (Ren et al., 2018) propose a meta-learning algorithm that learns weights corresponding to training examples based on their gradient directions. (Fang et al., 2020) learns dynamic importance weighting that iterates between weight estimation and weighted classification. weighting the instances for selective training (Ren et al., 2018; Fang et al., 2020). Recently, (Rizve et al., 2021) propose utilizing prediction uncertainty to perform label selection. All the above-mentioned methods require clean validation sets to infer parameters, whereas our method needs no clean annotated data. Inspired from the recent studies on memorization effects of DNNs that they learn clean data earlier than noisy data, we use learning order to select the samples.

Training dynamics. In deep learning regime, models with large capacity are typically more robust to outliers. Nevertheless, data examples can still exhibit diverse levels of difficulties. Arpit et al. (2017) finds that data examples are not learned equally when injecting noisy data into training. Easy examples are often learned first. Hacohen et al. (2019) furthers shows such order of learning examples is shared by different random initializations and neural architectures. Toneva et al. (2019) shows that certain examples are forgotten frequently during training, which means that they can be first classified correctly then incorrectly. Model performance can be largely maintained when removing those least forgettable examples from training.

Table 1: Noise ratios of different pseudo-label heuristics on NYT-Fine dataset.

Pseudo-label Heuristic	**Noise Ratio**
String-Match (Mekala et al., 2020)	31.80%
Contextualized String-Match (Mekala and Shang, 2020)	31.24%
Exclusive String-Match (Mekala et al., 2021)	52.13%
Clustering (Wang et al., 2020)	15.64%

3 Problem and Motivation

Weakly supervised classification refers to the problem with inputs (1) a set of unlabeled text documents $\mathcal{S} = \{x\}$, where $x \in \mathcal{X}$. (2) and M target labels $\mathcal{C} = \{1, \ldots, M\}$. Our goal is to find a labeling function $f : \mathcal{X} \to \mathcal{C}$ that maps every document x to its true label. Here we denote y^* as the *unknown* true label of a document x. To cold start the classification of unlabeled documents, a source of weak supervision has to be introduced, which can come from various sources such as label surface names (Wang et al., 2020), label-indicative seed words (Mekala and Shang, 2020), or rules (Karamanolakis et al., 2021a). Given a "weak" labeling function $w : \mathcal{X} \to \mathcal{C}$, pseudo-labels are then generated on a subset of the unlabeled documents, which yields a labeled subset $\mathcal{D} = \{(x, w(x))\}$. For convenience, we denote $\mathcal{D}[j]$ to be the set of all documents that are pseudo-labeled as class j in $\mathcal{D}$, namely $\mathcal{D}[j] = \{(x, w(x)) \in \mathcal{D} | w(x) = j\}$.
Pseudo-labels are noisy due to their heuristic nature. For example, as shown in Table 1, we consider NYT fine-grained dataset and generate pseudo-labels using four different strategies (Mekala and Shang, 2020; Mekala et al., 2020, 2021; Wang et al., 2020) and compute their noise ratios. We can observe that no strategy is perfect and all of them generate noisy labels, ranging from 15% to 50%.

When a classifier is trained on such noisy training data, it can make some high confident erroneous predictions. And, upon bootstrapping the classifier on unlabeled data, it has a snowball effect where such high confident erroneous predictions are added to the training data, and thus corrupting it more. As this process repeats for a few iterations, it adds more noise and significantly affects the final performance. Therefore, identifying and selecting the correctly labeled samples is necessary and has a huge potential for a boost in performance. Note that, if the labels are not selected carefully, it could instead hurt the performance.

Our pseudo-label selection problem. The weak supervision is likely to generate a noisy labeled set, which means $w(x) \neq y^*$ for some documents x. We denote $\mathcal{D}_{\checkmark}$ as the set of correctly labeled documents and $\mathcal{D}_{\times} = \mathcal{D} \setminus \mathcal{D}_{\checkmark}$ as the set of wrongly labeled documents, where $\mathcal{D}_{\checkmark} = \{(x, w(x)) | w(x) = y^*\}$. The problem of pseudo-label selection is thus to identify $\mathcal{D}_{\checkmark}$.

Note that pseudo-label selection is conceptually related to failure prediction (Hecker et al., 2018; Jiang et al., 2018a; Corbière et al., 2019) and out-of-distribution detection (Hendrycks and Gimpel, 2017; Devries and Taylor, 2018; Liang et al., 2018; Lee et al., 2018). However, the major difference here is for pseudo-label selection we have to detect wrong annotations in the training phase instead of inference phase.

4 Pseudo-label Selection methods

4.1 Confidence function-based selection

Confidence function $\kappa : \mathcal{X} \times \mathcal{C} \to [0, 1]$, assigns a value to each labeled document, which represents our confidence of its pseudo-label being correct. Then, we can perform the selection by choosing a threshold τ on confidence function. We denote the set of labeled documents selected based on κ and τ as $\hat{\mathcal{D}}_{\checkmark}(\kappa, \tau)$, namely

$$\hat{\mathcal{D}}_{\checkmark}(\kappa, \tau) = \{(x, w(x)) \in \mathcal{D} \mid \kappa(x, w(x)) > \tau\}$$

An optimal confidence function κ^* should be able to perfectly distinguish the correctly labeled documents from wrongly labeled ones, namely there exists a threshold τ^* such that $\hat{\mathcal{D}}_{\checkmark}(\kappa^*, \tau^*) = \mathcal{D}_{\checkmark}$.

Evaluation of a confidence function. Practical confidence functions may not be possible to suffice such ideal condition. There always exists a trade-off between *noise* $\epsilon(\kappa, \tau)$ and *coverage* $\phi(\kappa, \tau)$, defined as:

$$\epsilon(\kappa, \tau) = \frac{|\hat{\mathcal{D}}_{\checkmark}(\kappa, \tau) \cap \mathcal{D}_{\times}|}{|\hat{\mathcal{D}}_{\checkmark}(\kappa, \tau)|}, \quad \phi(\kappa, \tau) = \frac{|\hat{\mathcal{D}}_{\checkmark}(\kappa, \tau)|}{|\mathcal{D}|}.$$

The coverage is the fraction of labeled documents being selected and the noise is the fraction of wrongly labeled documents within selected documents. A small threshold leads to high coverage i.e. most labeled documents will be selected, thus being more noisy. And a high threshold leads to an opposite situation. Therefore, to evaluate a confidence function, we plot noise and coverage at various thresholds, which we refer as the *noise-coverage curve* (NC-curve) and compute the *area under the noise-coverage curve* (AUNC). As shown in figure 3, an optimal confidence function selects wrongly labeled documents only after selecting all the correctly labeled documents, hence generates a NC-curve in the shape of a rectifier, namely $\epsilon = \max(0, \phi - |\mathcal{D}_{\checkmark}|/|\mathcal{D}|)$. A random confidence function always selects the same fraction of wrongly labeled documents, hence generates a NC-curve with a constant value. An ideal confidence function should minimize AUNC.

Selection of the threshold. For a given confidence function, one wishes to select pseudo-labels based on a threshold such that the noise is low and the coverage is high. We define ratio between noise and coverage as *NC-ratio*, namely $r(\kappa, \tau) = \frac{\epsilon(\kappa,\tau)}{\phi(\kappa,\tau)}$. An optimal threshold has the lowest NC-ratio.

4.2 Learning order as a superior confidence function

In this section, we introduce learning order as confidence function and compare it with the commonly used probability score using previously mentioned evaluation metrics.

Probability score. One intuitive confidence function for pseudo-label selection is the model's prediction probability scores corresponding to the pseudo-labels. Specifically, let $\mathbf{f} : \mathcal{X} \to [0, 1]^{|\mathcal{C}|}$ be a probabilistic classifier trained on pseudo-labeled documents and $\mathbf{f}(x)[j]$ represents the predicted probability of document x belonging to class j, $\mathbf{f}(x)[w(x)]$ is used as the confidence function. However, due to the poor calibration of DNNs (Guo et al., 2017), probability scores of wrongly labeled documents are usually high. As a result, it might be difficult to distinguish correctly- and wrongly-labeled documents based on probability scores.

Learning order. Learning order of a pseudo-labeled document is the epoch when it is learnt during training, or more specifically when its label predicted by the model matches its given pseudo-label. Recent studies show that a DNN learns clean samples first and then gradually memorizes the noisy samples (Arpit et al., 2017). We thus hypoth-

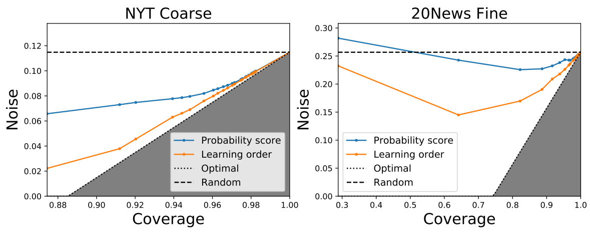


Figure 3: NC-curves of learning order and probability score with BERT as the classifier.

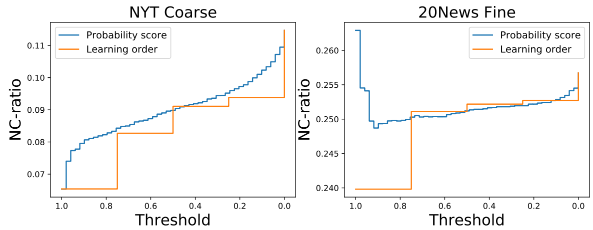


Figure 4: NC-ratios of learning order and probability score with BERT as the classifier.

esize that learning order can reflect the probability of wrong pseudo-label in terms of ranking.

We now utilize learning order to define a confidence function. Specifically, let $\mathbf{f}^t(\cdot)$ be the classifier being trained at epoch t, and T as the total number of epochs, the learning order of document x can be defined as

$$\eta(x, w(x)) = 1 - \frac{1}{T}\min\{t \mid \arg\max_j \mathbf{f}^t(x)[j] = w(x)\}, \quad (1)$$

where $t \in \{1, \dots, T\}$. Here we have negated and scaled the learning order to be complied with the convention of confidence function i.e. higher confidence implies higher probability of a correct label. We calculate the learning order at the granularity of epoch because the model would have seen all the training data by the end of an epoch, and hence, the learning order computed would be fair for all documents. In case when the epoch number is not sufficient to distinguish the documents, one can increase the granularity of the learning order, for example, the batch number at which the document is learnt. Granularity higher than the epoch incurs extra training cost as a document will be examined more than once in each epoch.

Learning Order vs Probability Score. We compare the effectiveness of learning order and probability scores as confidence functions. We plot NC-curves and NC-ratios of learning order and probability scores in figures 3, 4 on NYT-Coarse, 20News-Fine datasets. From figure 3, we observe that learning order has significantly smaller AUNC compared to the probability score. In easy datasets such as NYT-Coarse, it can even approach the optimal confidence function.

As shown in Figure 4, when selecting the optimal threshold, learning order has significantly lower NC-ratios for all datasets compared to probability score. Furthermore, the optimal thresholds of learning order for all datasets are almost the same. In contrast, the optimal thresholds of probability score vary greatly across different datasets due to the poor calibration of DNNs. Finally, we also observe that the NC-ratio for probability score often changes greatly around the optimal threshold, which poses difficulty in locating the optimal threshold. In contrast, since there are only few possible thresholds for learning order, it is easier to find the optimal threshold. Therefore, in terms of both performance and robustness, learning order is a more effective confidence function than probability score.

5 LOPS: Putting it all together

Motivated by previous analyses, we utilize learning order to select pseudo-labels. We train a classifier on all pseudo-labeled documents and track their first learnt epoch during training. The confidence function can then be calculated based on Equation (1). Finally, we rank the documents based on their confidence and select the top-$\tau\%$ for each label independently.

To maximize the efficiency of LOPS, we utilize the fact that the top-ranked documents are learned earlier, and conduct the confidence calculation and pseudo-label selection simultaneously during training. Specifically, for each label, a document is selected once it is learnt, until the fraction of selected documents exceeds $\tau\%$ in this label. Whenever the fractions of selected documents exceeds $\tau\%$ for all labels, we stop the training. The pseudo-code is shown in Algorithm 1. Note that LOPS can be plugged to any weakly-supervised classification framework as shown in Appendix A.1.

6 Experiments

In this section, we evaluate our label selection method on different state-of-the-art classifiers and weakly supervised text classification frameworks.

Algorithm 1: LOPS Method

Input: A set of documents $\mathcal{D}$ pseudo-labeled by w, Classifier $\mathbf{f}$.
Output: Selected documents $\hat{\mathcal{D}}_{\checkmark}$

```
for epoch t = 1, 2, ..., T do
    Train f on D
    for (x, w(x)) ∈ D do
        if argmax_j f(x)[j] = w(x) then
            if |D̂✓[w(x)]| / |D[w(x)]| < τ% then
                D̂✓ = D̂✓ ∪ {(x, w(x))}
    if |D̂✓[j]| / |D[j]| ≥ τ% for all j then
        Break
Return D̂✓
```

Table 2: Dataset statistics.

Dataset	# Docs	# labels	Avg Len
NYT-Coarse	13081	5	778
NYT-Fine	13081	26	778
20News-Coarse	17871	5	400
20News-Fine	17871	17	400
AGNews	120000	4	426
Books	33594	8	620

6.1 Datasets

We experiment on four datasets: New York Times (**NYT**), 20 Newsgroups (**20News**), **AGNews** (Zhang et al., 2015), **Books** (Wan and McAuley, 2018; Wan et al., 2019). NYT and 20News datasets also have fine-grained labels which are also used for evaluation. The dataset statistics are provided in Table 2 and more details are provided in Appendix A.2.

6.2 Compared Label Selection Methods

We compare with several metrics used for label selection mentioned below:

- **Probability:** We sort the prediction probabilities corresponding to pseudo-labels in descending order and select the same number of samples as LOPS in each iteration of bootstrapping.
- **Random:** We randomly select the same number of samples as LOPS in each iteration of bootstrapping. To avoid skewed selection, we sample in a stratified fashion based on class labels.
- **Learning Stability (stability):** (Dong et al., 2021) introduced a metric to measure the data quality based on the frequency of events that an example is predicted correctly throughout the training. We sort the samples based on learning stability in descending order i.e. most stable to least stable and select the same number of samples as LOPS in each iteration of bootstrapping.

We consider the same number of samples as LOPS in each iteration for all above baselines because we cannot tune individual thresholds for each dataset since there is no human-annotated data under the weakly supervised setting and one fixed threshold for all datasets doesn't work as the distribution of prediction probability varies across datasets. So, to perform controlled experiments with a fair comparison, we consider the same number of samples as LOPS in each iteration.

We also present experimental results without any label selection (denoted by ***No-Filter***) as lower bound and with all the wrongly annotated samples removed as ***Optimal***.

6.3 Experimental Settings

Seed Words. For all our experiments, we consider seed words used in (Mekala and Shang, 2020; Wang et al., 2020) as weak supervision and generate initial pseudo-labels using String-Match (Mekala et al., 2020) unless specified. The average number of seeds are 4 per class.

Text Classifiers. We experiment on four state-of-the-art text classifiers: (1) **BERT** (`bert-base-uncased`) (Devlin et al., 2018), (2) **RoBERTa** (`roberta-base`) (Liu et al., 2019), (3) **XLNet** (`xlnet-base-cased`) (Yang et al., 2019), and (4) **GPT-2** (Radford et al., 2019). We follow the same self-training method for all classifiers that starts with generating pseudo-labels, training a classifier on pseudo-labeled data, and bootstrap it on unlabelled data by adding samples whose prediction probabilities are greater than δ.

Following (Mekala and Shang, 2020), we assume that weak supervision $\mathcal{W}$ is of reasonable quality i.e. majority of pseudo-labels are good. Therefore, we set τ to 50%. While training the classifiers, we fine-tune RoBERTa for 3 epochs, BERT, XLNet, GPT-2 for 4 epochs. We bootstrap all the classifiers for 5 iterations with the probability threshold δ as 0.6.

Weakly Supervised Text Classification Frameworks. We experiment on state-of-the-art weakly supervised text classification methods: **ConWea** (Mekala and Shang, 2020), **X-Class** (Wang et al., 2020), **WeSTClass** (Meng et al., 2018), and **LOTClass** (Meng et al., 2020). Three of them are self-training-based methods and more details about these methods are mentioned in Appendix A.3.

6.4 Quantitative Results

We discuss the effectiveness of LOPS with different classifiers and weakly supervised text classification frameworks.

Table 3: Evaluation results on six datasets using different combinations of classifiers and pseudo-label selection methods. Initial pseudo-labels are generated using String-Match. Micro- and Macro-F1 scores and their respective standard deviations are presented in percentages. Abnormally high standard deviations are highlighted in *blue* and low performances are highlighted in *red*. Statistical significance results are in Appendix A.5

		Coarse-grained Datasts								Fine-grained Datasets			
		NYT-Coarse		20News-Coarse		AGNews		Books		NYT-Fine		20News-Fine	
Classifier	Method	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
BERT	No-Filter	90.1(0.17)	80.3(0.91)	77.3(0.27)	76.4(0.76)	75.4(0.64)	75.4(0.47)	55.7(0.54)	57.9(0.82)	77.2(0.36)	71.6(0.43)	70.0(0.30)	69.6(0.25)
	Random	90.3(0.47)	80.9(0.47)	79.0(1.00)	76.8(1.50)	76.3(0.35)	76.3(0.65)	56.1(0.18)	58.2(0.35)	78.4(0.94)	71.7(0.47)	71.4(0.50)	70.6(1.00)
	Probability	92.3(1.50)	85.1(2.00)	78.6(2.5)	77.5(3.00)	77.4(1.25)	77.6(1.34)	54.3(1.12)	56.5(1.43)	46.6(2.50)	22.3(0.50)	47.8(23.50)	47.9(23.50)
	Stability	93.3(0.50)	86.5(0.50)	76.7(5.00)	75.4(5.00)	79.3(0.75)	**79.5(0.35)**	55.0(0.43)	57.0(0.19)	48.1(29.50)	35.5(33.50)	73.5(0.50)	72.5(1.00)
	LOPS	**94.6(0.36)**	**88.4(0.50)**	**81.7(1.00)**	**80.7(0.43)**	**79.5(0.86)**	**79.5(0.58)**	**57.7(0.87)**	**59.5(0.46)**	**84.3(0.54)**	**81.6(0.34)**	**73.8(0.61)**	**72.7(1.00)**
	Optimal	98.3(0.27)	96.4(0.37)	94.7(0.37)	94.9(0.61)	89.4(0.46)	89.3(0.76)	76.2(0.21)	76.7(0.19)	97.4(0.71)	92.2(0.62)	87.6(0.37)	86.5(0.36)
RoBERTa	No-Filter	90.2(0.41)	82.1(0.24)	76.5(0.41)	75.7(0.58)	74.4(0.44)	74.2(0.71)	57.6(0.29)	58.6(0.53)	79.4(0.65)	76.6(0.54)	67.4(0.67)	67.3(0.87)
	Random	92.3(0.21)	84.4(0.82)	76.5(1.00)	74.5(1.00)	74.6(0.32)	74.2(0.27)	56.4(0.57)	58.7(0.32)	76.6(1.25)	74.8(0.34)	68.4(0.23)	68.5(0.23)
	Probability	**93.4(0.48)**	**87.5(1.00)**	76.7(0.50)	75.4(1.00)	76.2(0.89)	76.3(1.12)	56.2(1.28)	57.4(1.85)	26.6(23.00)	14.4(11.50)	46.2(23.00)	45.3(23.50)
	Stability	90.5(1.09)	83.3(0.50)	**78.5(1.00)**	**76.0(1.50)**	**76.5(0.48)**	**76.5(0.64)**	58.5(1.18)	59.5(1.06)	21.5(12.50)	9.2(5.00)	70.3(1.00)	70.6(1.00)
	LOPS	92.4(2.99)	85.6(3.00)	77.5(2.00)	75.8(2.00)	75.6(0.22)	75.5(0.27)	**59.7(0.41)**	**60.5(0.45)**	**81.8(0.90)**	**80.7(0.50)**	**70.7(0.68)**	**70.8(0.34)**
	Optimal	98.2(0.17)	96.1(0.16)	94.3(0.74)	94.5(0.35)	89.7(0.17)	89.3(0.28)	76.5(0.29)	77.7(0.22)	97.4(0.34)	92.8(0.26)	85.3(0.32)	85.5(0.65)
XLNet	No-Filter	89.2(0.74)	80.1(0.64)	77.6(0.39)	75.4(0.68)	72.7(0.97)	72.4(0.53)	57.6(0.31)	58.7(0.46)	77.4(0.34)	71.3(0.75)	60.7(0.74)	66.5(0.61)
	Random	90.7(0.03)	80.5(0.51)	78.6(0.50)	75.4(1.00)	67.5(0.22)	67.4(0.63)	57.5(0.43)	58.3(0.45)	76.6(0.94)	72.7(0.70)	67.3(0.49)	67.2(0.32)
	Probability	91.3(0.29)	**83.4(0.50)**	77.4(1.00)	75.2(0.30)	70.1(1.09)	70.4(1.14)	54.6(1.42)	56.3(1.26)	38.2(6.50)	36.5(1.00)	69.5(0.82)	69.2(0.12)
	Stability	**91.4(1.00)**	82.3(1.5)	79.7(1.50)	77.6(1.50)	74.3(1.10)	74.5(0.87)	56.3(0.88)	58.1(0.97)	79.5(0.50)	76.3(1.10)	68.5(0.49)	68.4(1.00)
	LOPS	89.5(0.17)	81.4(0.90)	**82.5(0.50)**	**81.2(0.2)**	**77.7(0.57)**	**77.7(0.54)**	**58.5(0.65)**	**59.4(0.67)**	**80.7(0.22)**	**77.4(0.83)**	**70.6(0.31)**	**70.4(0.27)**
	Optimal	98.3(0.12)	96.5(0.21)	94.5(0.23)	94.4(0.29)	89.3(0.28)	89.7(0.39)	76.4(0.44)	76.3(0.43)	97.4(0.32)	93.6(0.38)	86.6(0.43)	86.4(0.35)
GPT-2	No-Filter	91.1(0.24)	82.3(0.28)	78.4(0.26)	76.3(0.38)	61.3(0.28)	61.2(0.43)	51.6(0.41)	53.3(0.37)	76.2(0.41)	69.5(0.38)	70.5(0.46)	70.4(0.38)
	Random	90.2(0.42)	80.2(0.56)	79.7(0.46)	78.4(0.32)	68.2(0.18)	68.1(0.19)	53.4(0.46)	55.3(0.42)	77.5(0.52)	70.4(1.02)	69.4(0.21)	69.3(0.29)
	Probability	93.3(1.04)	85.5(1.13)	80.4(1.49)	78.5(1.50)	66.2(0.69)	66.6(0.89)	51.7(1.11)	54.5(1.09)	76.7(0.57)	71.3(0.69)	69.4(1.21)	69.3(1.18)
	Stability	94.4(0.56)	88.6(0.59)	81.4(1.02)	78.6(1.50)	72.4(0.58)	72.3(0.53)	53.6(1.02)	55.3(1.13)	79.4(0.62)	75.3(0.65)	**70.6(0.68)**	70.4(0.63)
	LOPS	**95.2(0.49)**	**89.1(0.51)**	**82.5(0.57)**	**80.3(0.63)**	**75.7(0.52)**	**75.3(0.31)**	**56.8(0.89)**	**58.6(0.63)**	**80.4(0.09)**	**76.3(0.21)**	**70.6(0.76)**	**70.5(0.48)**
	Optimal	98.3(0.24)	96.2(0.21)	94.2(0.23)	93.3(0.27)	88.7(0.26)	88.4(0.28)	72.3(0.19)	73.7(0.22)	97.3(0.18)	92.4(0.19)	86.1(0.35)	85.5(0.38)

Table 4: Evaluation results of weakly supervised text classification frameworks with LOPS. This demonstrates that LOPS can be easily plugged in and improves the performance.

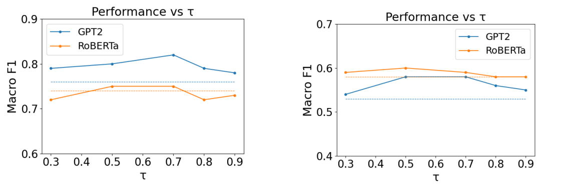
	NYT-Coarse		NYT-Fine		20News-Coarse		20News-Fine		AGNews		Books	
Method	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1	Mi-F1	Ma-F1
	ConWea											
No-Filter	93.1	87.2	87.4	77.4	74.3	74.6	68.7	68.7	73.4	73.4	52.3	52.6
LOPS	**94.2**	**90.1**	**87.5**	**78.6**	**79.7**	**78.4**	**70.4**	**70.6**	**79.2**	**79.2**	**57.5**	**58.7**
	X-Class											
No-Filter	**96.3**	**93.3**	86.6	**74.7**	58.2	61.1	70.4	70.4	82.4	82.3	53.6	54.2
LOPS	96.2	**93.3**	**86.8**	73.8	**60.7**	**62.3**	**71.2**	**71.2**	**83.6**	**82.7**	**54.2**	**56.3**
	WeSTClass											
No-Filter	92.3	86.0	67.1	60.4	53.2	49.4	54.9	54.9	80.4	80.1	49.7	48.1
LOPS	**93.4**	**88.1**	**68.4**	**63.8**	**53.3**	**51.5**	**61.1**	**60.5**	**81.4**	**81.3**	**51.2**	**49.8**
	LOTClass											
No-Filter	**70.1**	**30.3**	**5.3**	**4.1**	**47.0**	**35.0**	**12.3**	**10.6**	84.9	84.7	**19.9**	**16.1**
LOPS	**70.1**	**30.3**	3.5	2.9	45.7	32.6	7.8	4.1	**86.2**	**86.1**	15.8	10.3

6.4.1 Results: Different Classifiers

We summarize the evaluation results with different combinations of classifiers and selection methods in Table 3. All experiments are run on three random seeds and mean, standard deviations are reported in percentages.

As shown in Table 3, upon plugging our proposed method LOPS, we observe a significant boost in performance over No-Filter with all the classifiers. We observe that LOPS always outperforms random selection which shows that the selection in LOPS is strategic and principled. LOPS performs better than probability and stability based selection methods in most of the cases. This shows that LOPS is very effective in removing wrongly labeled and preserving correctly labeled samples.

We also observe a significant boost in performance over No-Filter with all the classifiers in the case of fine-grained datasets as well. In some cases like BERT on NYT-Fine, the improvement is as high as 7 points on micro-f1 and 10 points on macro-f1. We observe abnormally low performances of probability and stability based selection methods in some scenarios (highlighted in *red*). This is because the number of noisy labels are more in fine-grained datasets and gets amplified with self-training and resulting in high noise. And also, based on Figure 3, the calibration is so poor on fine-grained datasets that the probability score is even worse than random. Moreover, we also observe that probability and stability based selections are biased towards majority labels and select wrong majority labels over correct minority labels. For example, the precision of pseudo-labels belonging to minority classes like *cosmos*, *gun control*, and *abortion* in NYT-Fine before selection is 100% and it selected almost none of these whereas it selected 700 wrong documents belonging to a majority label like, *international business*. Although stratified selection can be employed to address this problem, this ends up having a same threshold and selecting



(a) 20News Coarse (b) Books

Figure 5: Macro-F$_1$ scores w.r.t. τ on Books & 20News-Coarse datasets using GPT2 and RoBERTa as classifiers with LOPS. The dashed lines represent performance with no label selection.

a fixed ratio of samples for every dataset, which might not be optimal for every dataset as shown in Figure 4. We have to note unusually high standard deviation for probability and stability based selection methods in some cases (highlighted in *blue*) as observed in Figure 4.

This demonstrates that these selection methods are unstable. LOPS is comparatively more stable and its effectiveness is largely due to its invariance. Although probability and stability based selection methods outperform LOPS in a few cases, their unstable nature makes them unreliable. Therefore, we believe LOPS is a superior method than compared selection methods.

We observe that LOPS uplifts the performance quite close to supervised methods. This demonstrates that LOPS acts as an effective plugin and helps in closing the performance gap between the weakly supervised and supervised settings.

6.4.2 Results: Different Weakly-Supervised Text Classification Methods

We summarize the evaluation results with different weakly supervised methods in Table 4. The results demonstrate that LOPS improves the performance of ConWea and WeSTClass significantly on all datasets and X-Class sometimes. Note that, X-Class sets a confidence threshold and selects only top-50% instances, which provides a hidden advantage and LOPS improves the performance on top of it for some datasets. We have to note the significantly low performance of LOTClass. It is observed that LOTClass requires a wide variety of contexts of label surface names from the input corpus to generate high quality category vocabulary, which plays a key role in performance (Wang et al., 2020). The performance is comparitively worse in fine-grained classes than coarse-grained classes because LOTClass assumes that the replacements of label surface names are indicative of its respective label. However, this might not be a valid assumption for fine-grained classes (Mekala et al., 2021). Among the datasets we experimented on, these requirements are satisfied only by AGNews dataset where there are many documents(120000) classified broadly into 4 categories and we observe a performance boost using LOPS on this dataset. Due to poor quality of pseudo-labels for other datasets, there is no increment in performance with LOPS.

6.5 Performance vs τ

We conduct experiments to study the effect of τ on performance. The plot of macro-f1 score vs τ on 20News-coarse and Books dataset using GPT2 and RoBERTa classifiers is shown in Figure 5. We observe that the performance increases initially and gradually drops down at higher τ values. The lower τ values imply being highly selective and thus the few number of selected samples are not enough for the model to generalize. The higher τ values imply poor selection with many noisy labels, making the performance to drop. From the plot, we can observe that the performance is robust for middle τ values i.e. $50 - 70\%$.

6.6 Example samples

A few incorrectly pseudo-labeled samples from NYT-Fine dataset that are selected by probability-based selection by RoBERTa are shown in Table 6 in Appendix A.6. We observe a high probability assigned to each incorrect pseudo-label whereas these are learnt by the classifier at later epochs. These wrongly annotated samples induce error that gets propagated and amplified over the iterations. By not selecting these wrong instances, LOPS curbs this and boosts the performance.

7 Conclusion and Future Work

In this paper, we proposed LOPS, a novel learning order inspired pseudo-label selection method. Our method is inspired from recent studies on memorization effects that showed that clean samples are learnt first and then wrong samples are memorized. Experimental results demonstrate that our method is effective, stable and can act as a performance boost plugin on many text classifiers and weakly supervised text classification methods. It outperforms several label selection methods based on probability and learning stability. In the future, we are interested in analyzing the role of noise and investigate any positive consequences of noise in text classification.

8 Ethical Consideration

This paper proposes a label selection method for weakly supervised text classification frameworks. The aim of the paper is to detect the noise caused by the heuristic pseudo-labels and we don't intend to introduce any biased selection. Based on our experiments, we manually inspected some filtered samples and we didn't find any underlying pattern. Hence, we do not anticipate any major ethical concerns.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Chidubem Arachie and Bert Huang. 2021. Constrained labeling for weakly supervised learning. In *Uncertainty in Artificial Intelligence*, pages 236–246. PMLR.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. 2019. Addressing failure prediction by learning model confidence. In *NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Terrance Devries and Graham W. Taylor. 2018. Learning confidence for out-of-distribution detection in neural networks. *ArXiv*, abs/1802.04865.

Chengyu Dong, Liyuan Liu, and Jingbo Shang. 2021. Data profiling for adversarial training: On the ruin of problematic data. *CoRR*, abs/2102.07437.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. 2020. Rethinking importance weighting for deep learning under distribution shift. *arXiv preprint arXiv:2006.04662*.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. 2018. Bias-reduced uncertainty estimation for deep neural classifiers. *arXiv preprint arXiv:1805.08206*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Guy Hacohen, Leshem Choshen, and D. Weinshall. 2019. Let's agree to agree: Neural networks share classification order on real datasets. *arXiv: Learning*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*.

Simon Hecker, Dengxin Dai, and Luc Van Gool. 2018. Failure prediction for autonomous driving. *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1792–1799.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ArXiv*, abs/1610.02136.

Heinrich Jiang, Been Kim, and Maya R. Gupta. 2018a. To trust or not to trust a classifier. In *NeurIPS*.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018b. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR.

Giannis Karamanolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah. 2021a. Self-training with weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 845–863, Online. Association for Computational Linguistics.

Giannis Karamanolakis, Subhabrata (Subho) Mukherjee, Guoqing Zheng, and Ahmed H. Awadallah. 2021b. Self-training with weak supervision. In *NAACL 2021*. NAACL 2021.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv: Learning*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". *arXiv preprint arXiv:1706.02613*.

Dheeraj Mekala, Varun Gangal, and Jingbo Shang. 2021. Coarse2fine: Fine-grained text classification on coarsely-grained annotated data. *arXiv preprint arXiv:2109.10856*.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333.

Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. 2020. Meta: Metadata-empowered weak supervision for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8351–8361.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992. ACM.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343. PMLR.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 25–32. Association for Computational Linguistics.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*.

Fangbo Tao, Chao Zhang, Xiusi Chen, Meng Jiang, Tim Hanratty, Lance Kaplan, and Jiawei Han. 2015. Doc2cube: Automated document allocation to text cube via dimension-aware joint embedding. *Dimension*, 2016:2017.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and G. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. *ArXiv*, abs/1812.05159.

Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2020. X-class: Text classification with extremely weak supervision. *arXiv preprint arXiv:2010.12794*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

A Appendix

A.1 LOPS for weakly supervised classification methods

The pseudo code for self-training with LOPS is shown in Algorithm 2.

Algorithm 2: Self-training with LOPS label selection

Input: Unlabeled data $\mathcal{D}$, Classifier C, Weak Supervision $\mathcal{W}$.
Output: Prediction labels $predLabs$
$\hat{\mathcal{D}}$ = Generate Pseudo-labels from $\mathcal{D}$, $\mathcal{W}$
for $it \in \{1, 2, 3, \ldots, n_{its}\}$ **do**
    $\mathcal{D}_{sel}$ = LOPS ($\hat{\mathcal{D}}$, C)
    Train C on $\mathcal{D}_{sel}$
    $predLabs, predProbs$ = Predict(C, $\mathcal{D}$)
    $\hat{\mathcal{D}} = \hat{\mathcal{D}} \cup \{x \mid predProbs(x) > \delta\}$
Return $predLabs$

A.2 Datasets

The details of datasets are provided below:

- **The New York Times (NYT):** The NYT dataset is a collection of news articles published by The New York Times. They are classified into 5 coarse-grained genres (e.g., science, sports) and 25 fine-grained categories (e.g., music, football, dance, basketball).
- **The 20 Newsgroups (20News):** The 20News dataset[2] is a collection of newsgroup documents partitioned widely into 6 groups (e.g., recreation, computers) and 20 fine-grained classes (e.g., graphics, windows, baseball, hockey). Following (Wang et al., 2020), coarse- and fine-grained miscellaneous labels are ignored.
- **AGNews** (Zhang et al., 2015) is a huge collection of news articles categorized into four coarse-grained topics such as business, politics, sports, and technology.
- **Books** (Wan and McAuley, 2018; Wan et al., 2019) is a dataset containing description of books, user-book interactions, and users' book reviews collected from a popular online book review website Goodreads[3]. Following (Mekala et al., 2020), we select books belonging to eight popular genres. Using the title and description as text, we aim to predict the genre of a book.

A.3 Compared Weakly Supervised Text Classification Methods

We compared with following state-of-the-art weakly supervised text classification methods described below[4]:

- **ConWea** (Mekala and Shang, 2020) is a seed-word driven iterative framework that uses pre-trained language models to contextualize the weak supervision.
- **X-Class** (Wang et al., 2020) takes only label surface names as supervision and learns class-oriented document representations. These document representations are aligned to classes, computing pseudo labels for training a classifier.
- **WeSTClass** (Meng et al., 2018) generates pseudo documents using seed information and refines the model through a self-training module that bootstraps on unlabeled documents.
- **LOTClass** (Meng et al., 2020) queries replacements of class names using BERT (Devlin et al., 2018) and constructs a category vocabulary for each class. This is used to pseudo-label the documents via string matching. A classifier is trained on this pseudo-labeled data with further self-training.

We use the public implementations of these methods and modify them to plug-in our filter. Specifically, in WeSTClass and LOTClass, we add our filter after generating the pseudo documents; in ConWea, we add our filter before training the text classifier; and for X-Class, we plug-in our filter after learning the document-class alignment.

A.4 Experimental Settings

Train-Test sets. We remove the labels in the whole dataset and our task is to assign labels to these unlabeled samples. We measure our performance on the whole dataset by comparing it with their respective gold labels.
Computation Infrastructure. We performed our experiments on NVIDIA RTX A6000 GPU. The batch size for training BERT is 32, RoBERTa is 32, GPT2 is 4, XLNet is 1. The running time for BERT and RoBERTa took 3 hrs, GPT2 took 6 hours, and XLNet took 12 hrs.

A.5 Statistical Significance Tests

We perform a paired t-test between LOPS and each of the other baseline filtering techniques for all clas-

[2] http://qwone.com/~jason/20Newsgroups/
[3] https://www.goodreads.com/

[4] We also considered experimenting on ASTRA, however the instructions to run on custom datasets were not made public yet.

sifiers and on all datasets. The results are showed in Table 5. From these p-values, we can conclude that the performance improvement over baselines is significant.

A.6 Example samples

A few incorrectly pseudo-labeled samples from NYT-Fine dataset that are selected by probability-based selection with RoBERTa as classifier are shown in Table 6.

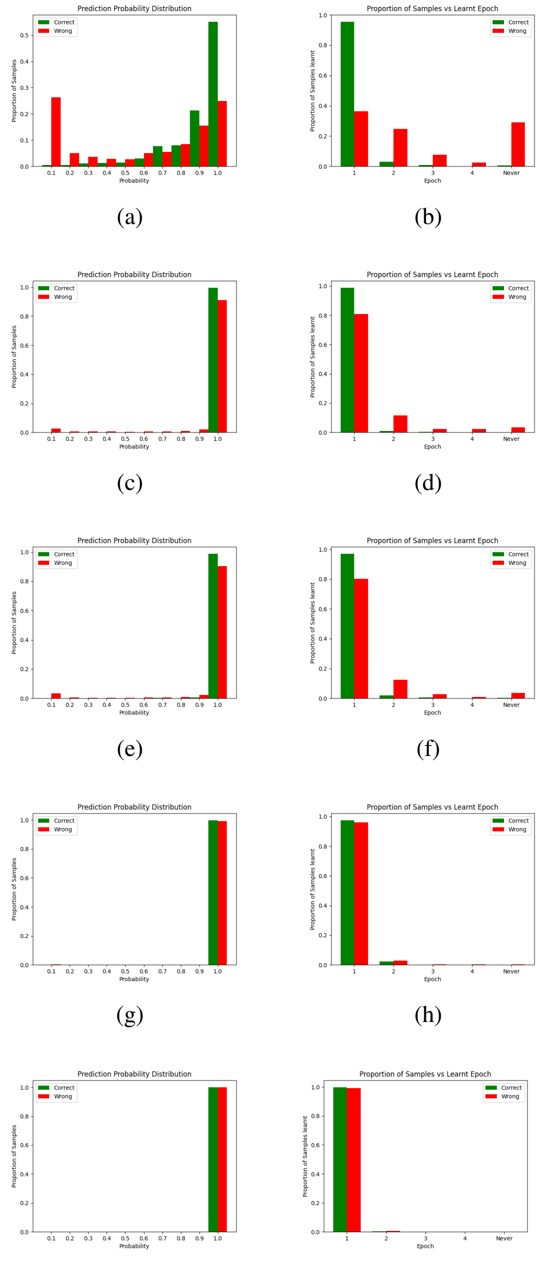


Figure 6: Distributions of correctly and wrongly labeled pseudo-labels using different selection strategies on all datasets for its initial pseudo-labels. The base classifier is BERT. Each row represents a dataset. Figure (a), (b) represents NYT-Fine, (c), (d) represents 20News-Coarse, (e), (f) represents 20News-Fine, (g), (h) represents Books, and (i), (j) represents AGNews datasets respectively. Left column is based on the softmax probability of samples' pseudo-labels and right column is based on the earliest epochs at which samples are learnt.

Table 5: Statistical significance results.

Classifier	Method	NYT-Coarse	NYT-Fine	20News-Coarse	20News-Fine	AGNews	Books
BERT	No-Filter	1.93×10^{-112}	1.92×10^{-105}	7.08×10^{-80}	9.37×10^{-79}	1.05×10^{-74}	7.15×10^{-96}
	Random	1.58×10^{-115}	2.01×10^{-105}	5.98×10^{-94}	7.32×10^{-39}	4.26×10^{-81}	3.25×10^{-100}
	Probability	1.69×10^{-112}	6.25×10^{-189}	4.19×10^{-120}	6.71×10^{-136}	5.13×10^{-71}	8.72×10^{-123}
	Stability	2.63×10^{-33}	2.41×10^{-194}	2.78×10^{-58}	4.07×10^{-9}	1.36×10^{-45}	1.24×10^{-97}
RoBERTa	No-Filter	6.06×10^{-100}	1.82×10^{-63}	5.4×10^{-3}	3.09×10^{-109}	2.13×10^{-57}	1.15×10^{-22}
	Random	8.38×10^{-94}	3.55×10^{-71}	3.26×10^{-39}	5.20×10^{-101}	5.12×10^{-72}	1.75×10^{-61}
	Probability	5.27×10^{-62}	9.18×10^{-193}	1.39×10^{-71}	1.13×10^{-85}	4.03×10^{-24}	2.16×10^{-72}
	Stability	1.46×10^{-86}	3.39×10^{-188}	6.28×10^{-5}	8.71×10^{-107}	1.17×10^{-76}	1.81×10^{-65}
XLNet	No-Filter	3.14×10^{-79}	4.68×10^{-139}	5.42×10^{-112}	4.17×10^{-103}	1.69×10^{-114}	5.63×10^{-107}
	Random	3.26×10^{-71}	2.97×10^{-48}	2.56×10^{-77}	5.32×10^{-75}	6.38×10^{-32}	4.38×10^{-48}
	Probability	4.12×10^{-29}	1.36×10^{-63}	7.25×10^{-19}	6.27×10^{-47}	1.57×10^{-31}	6.23×10^{-32}
	Stability	6.17×10^{-29}	4.27×10^{-44}	1.47×10^{-73}	3.57×10^{-41}	1.79×10^{-28}	3.48×10^{-56}
GPT-2	No-Filter	6.09×10^{-50}	1.10×10^{-98}	2.05×10^{-57}	1.22×10^{-5}	4.68×10^{-91}	1.56×10^{-65}
	Random	2.54×10^{-22}	6.97×10^{-81}	4.25×10^{-91}	9.89×10^{-38}	6.39×10^{-77}	8.70×10^{-63}
	Probability	5.52×10^{-49}	2.37×10^{-89}	7.02×10^{-85}	1.05×10^{-83}	1.99×10^{-63}	3.44×10^{-49}
	Stability	6.15×10^{-110}	3.88×10^{-31}	3.40×10^{-66}	6.27×10^{-78}	2.21×10^{-47}	2.36×10^{-41}

Table 6: Incorrectly pseudo-labeled samples selected by probability-based selection are shown below. These samples are learnt at later epochs, thus LOPS avoids selecting them.

Document	**Pseudo-label**
Corinthians have received offer from tottenham hotspur for brazil's paulinho although the midfielder said on saturday he would not decide his future until after the confederations cup ."there is an official offer from tottenham to corinthians but, as i did when there was an inter milan offer, i'll sit and decide with my family before i make any decision," paulinho told reporters.	Football Softmax Prob: 0.96 Learnt Epoch: 2
Brittney griner and elena delle donne were poised to make history as the first pair of rookies from same class to start wnba all-star game. Now, neither will be playing as both are sidelined with injuries. It's a tough blow for the league, which has been marketing the two budding stars.	Baseball Softmax Prob: 0.96 Learnt Epoch: 2
Denmark central defender simon kjaer has joined french side lille from vfl wolfsburg on a four-year deal. Lille paid two million euros. 72 million pounds for the 24-year-old kjaer, who has won 35 caps for his country. He joined wolfsburg from palermo for 12 million euros.	Intl. Business Softmax Prob: 0.94 Learnt Epoch: 2
Fiorentina striker giuseppe rossi is quickly making up for lost time after suffering successive knee ligament injuries which kept him out of action for the best part of two years.	Football Softmax Prob: 0.95 Learnt Epoch: 2